In the Specification:

Please amend the Specification as shown below. Applicant respectfully submits that the proposed amendments are to correct various informalities in the Specification, and that no new matter is being added.

Please replace paragraph **[0002]** with new paragraph **[0002]**, shown below.

[0002]   This application claims priority from provisional application "SYSTEM FOR SOFTWARE APPLICATION DEVELOPMENT AND MODELING," Application No. 60/238,561, filed October 4, 2000, and "SYSTEM AND METHOD FOR COMPUTER CODE GENERATION", Application No. 60/238,559, filed October 4, 2000, and is related to "SYSTEM AND METHOD FOR COMPUTER CODE GENERATION", Application No. 09/970,741, Inventors Todd Little, Loren Konkus, Gilles Lavalou and Timo Metsaportti, filed October 4, 2001, all of which are incorporated herein by reference.

Please replace paragraph **[0016]** with new paragraph **[0016]**, shown below.

[0016]   Roughly described, the invention provides an environment that incorporates a Rational Rose compatible UML design tool within a software application design and development product, to comprise an integrated software application, development, and modeling system. One embodiment of the invention can incorporate a software code Generator Framework to automatically generate code related to the design. This software code Generator Framework is described in further detail in co-pending application "SYSTEM AND METHOD FOR COMPUTER CODE GENERATION", Application No. 09/970,741, Inventors Todd Little, Loren Konkus, Gilles Lavalou and Timo Metsaportti, filed October 4, 2001, incorporated herein. It provides a common set of standards and application programming interfaces (APIs) to generate code and configuration files from any data source. A primary goal in developing the Generator Framework is to unify the code generation

- 2 -

techniques implemented in the Builder family of products, by introducing sufficient abstraction levels. Built-in (or generic) rules are introduced in the generator framework. A data navigation layer isolates the generator framework from the data sources used. Filters can be added to the framework to transform data. Notifiers are used by the generator framework to notify external components about the generation process.

Please insert after paragraph **[0018]** new paragraph **[0018.1]**, shown below.

[0018.1] As used herein the term Java™ is a trademark of Sun Microsystems, Inc. The terms Windows™, ActiveX™, and SQL Server™ are trademarks of Microsoft Corporation. The terms Rational Rose™, Rose™, Rose 98 Enterprise Edition™, Informix™, and DB2™ are trademarks of International Business Machines (IBM) Corporation. The term RogueWave™ is a trademark of Rogue Wave Software, Inc. The terms WebLogic™, WebLogic Server™, and TUXEDO™ are trademarks of BEA Software, Inc. The term CORBA™ is a trademark of the Object Management Group. The term Oracle™ is a trademark of Oracle Corporation, The term Sybase™ is a trademark of Sybase, Inc.

Please replace paragraph **[0065]** with new paragraph **[0065]**, shown below.

[0065] **Figure 1** shows an overview of a software development system in accordance with the invention. The software development system or product includes functions and features necessary to support the design, verification, and generation of M3 server applications. As shown in Figure 1, the complete development system 102 includes the Expert System 104 itself, and a version of the Rational Rose designer 106. The Expert System can be provided as a plugin into a Builder Generator Framework, as described more fully in co-pending application "SYSTEM AND METHOD FOR COMPUTER CODE GENERATION", Application No. 09/970,741, Inventors Todd Little, Loren Konkus, Gilles Lavalou and Timo Metsaportti, filed October 4, 2001, incorporated herein.

The Rational Rose designer can be used to reverse engineer input database (RDBMS) files 108 and input IDL files 110, for subsequent use in generating servers or applications. A UML model 112 is used to define the server model under design. When the design is completed to the satisfaction of the software designer or developer, the developer can generate the necessary output IDL files 114, which are then stored in an interface repository 116 for later use. A test client 118 or client application, and a server 120 or server application can also be generated. Together these generated client and server applications are used to comprise an integrated design or development environment (IDE) 102.